Gawarit-Gawarit - instrukcja użytkownika

Kacper Skelnik, Wojciech Tyczyński
Politechnika Warszawska, Wydział Fizyki

24 czerwca 2020

Streszczenie

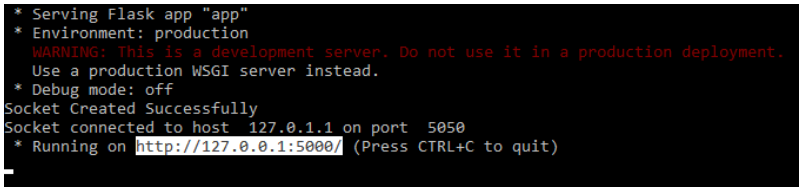
Instrukcja użytkownika do obsługi chatu działającego jako aplikacja webowa, opartego na protokole TCP. Aplikacja łączy się z serwerem tylko za pomocą tradycyjnych socketów. Back-end jest wykonany w frameworku Flask (z drobną pomocą skryptu Bashowego). Front-end łączy w sobie HTML, CSS i JavaScript. Całość jest połączona za pomocą języka Python. Zawsze aktualna wersja programu znajduję się pod linkiem

1 Opis aplikacji

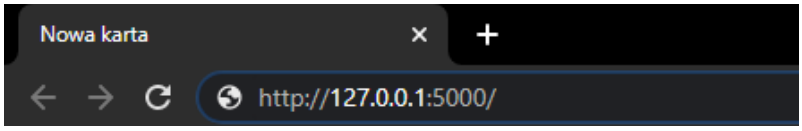
Program pozwala na pisanie do znajomych w czasie rzeczywistym. Po włączeniu odpowiedniego skryptu, aplikacje można uruchomić w przeglądarce. Tam znajdują się wszystkie funkcjonalności omówione w specyfikacji (z wyjątkiem wysyłania emotikon). Poniżej zostaną przedstawione szczegóły obsługi każdej z nich, oraz krótki raport z testowania aplikacji. Zawsze aktualna wersja oprogramowania znajduję się pod linkiem

2 Instalacja wszystkich niezbędnych komponentów, bibliotek i uruchomienie aplikacji

Do uruchomienia aplikacji wymagany jest Python w wersji co najmniej 3.6.9 wraz z zainstalowanym pip. Przed pierwszym uruchomieniem aplikacji należy uruchomić skrypt setup.sh który jest odpowiedzialny za stworzenie środowiska dla aplikacji. Skrypt sprawdza czy zainstalowany jest pakiet virtualenv do tworzenia wirtualnych środowisk pythona. W przypadku gdy pakiet nie jest zainstalowany następuje pobranie i instalacja pakietu. Przy pomocy pakietu tworzone i aktywowane jest wirtualne środowisko do którego następuje pobranie i zainstalowanie wszystkich niezbędnych bibliotek i pakietów. Lista pakietów i bibliotek oraz ich wymaganych wersji znajduje się w pliku requirements.txt.
Po przygotowaniu środowiska należy uruchomić skrypt run_client.sh który odpowiedzialny jest za uruchomienie aplikacji. Po uruchomieniu skryptu, uruchomi się aplikacja i poprosi o wprowadzenie adresu ip servera do którego chcemy się podłączyć. Po wprowadzeniu poprawnego adresu ip w terminalu wyświetli się link do aplikacji.



Link należy skopiować i wkleić w przeglądarce internetowej, a następnie uruchomić.



3 Rejestracja

Po odpaleniu aplikacji w przeglądarce użytkownik zobaczy stronę do rejestracji. Znajdujące się tam pola są zaprogramowane tak, że sprawdzają czy nazwa użytkownika ma długość od 4 do 25 znaków, oraz czy podane hasła są takie same (oczywiście są to pola wymagane). Jeżeli spełniają te warunki dane są wysyłane do serwera (po użyciu przycisku żegister"), który sprawdza czy użytkownik o takiej nazwie już nie istnieje. Jeżeli istnieje aplikacja wyświetli odpowiedni komunikat. Po udanej rejestracji następuję przeniesienie użyt-

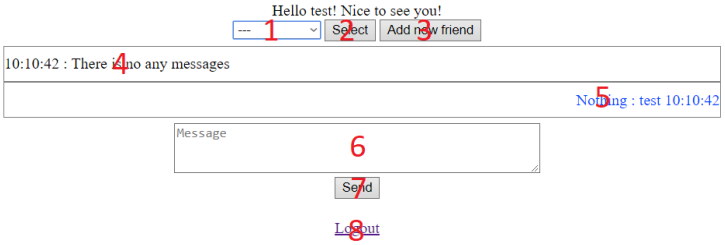
kownika do stronty z możliwością logowania.

4 Logowanie

Analogicznie jak przy rejestracji program wysyła dane podane przez użytkownika do serwera, który sprawdza czy zgadzają się one z tymi w bazie danych. Jeżeli się zgadzają serwer odsyła stosowną wiadomość i następuje przeniesienie do strony z chatem.

5 Chat

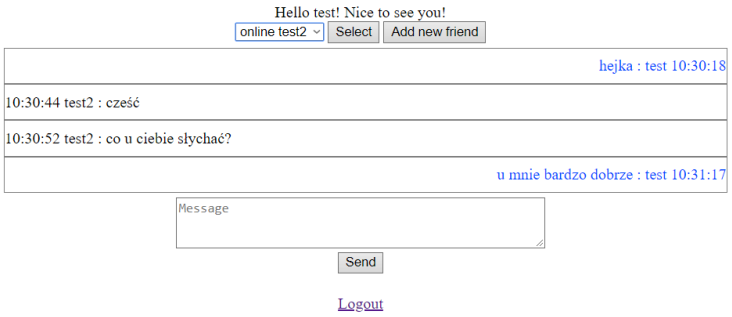
Po przejściu do okna czatu użytkownik widzi następujący obraz.



Opis elementów:

1. Lista znajomych (domyślnie ustawiona na ”puste pole” tak aby nowy użytkownik bez znajomych od razu widział chat)
2. Przycisk zatwierdzenia wyboru znajomego. Po wybraniu go z listy należy dodatkowo potwierdzić wybór aby strona załadowała wiadomości z wybranym znajomym.
3. Przycisk przenoszący do strony z możliwością dodania znajomego
4. Wiadomości od znajomego
5. Wiadomości do znajomego
6. Pole umożliwiające wpisanie wiadomości
7. Przycisk wysyłający wiadomość
8. Wylogowanie się z aplikacji (po naciśnięciu należy jeszcze raz włączyć aplikację, aby ta mogła na nowo połączyć się z serwerem)

Po wybraniu znajomego nastąpi przeniesienie do chatu z widocznymi wiadomościami historycznymi, oraz informacją czy ten jest online



Jeżeli drugi użytkownik zaloguje się w momencie w którym mamy go wybranego, aby zobaczyć zmiane statusu niezbędne będzie odświeżenie strony poprzez naciśnięcie jeszcze raz przycisku ”Select”, wysłanie nowej wiadomości lub zrestartowanie strony. Z powodów praktycznych w oknie czatu widać tylko 12 najnowszych wiadomości (z czego max 6 własnych). Wiadomości historyczne jednak są w w bazie danych. Do usprawnienia tej funkcjonalności można dodać przesuwany chat. Z racji, że chat działa na zasadzie protokołu w którym bardzo istotna jest kolejność wysyłanych danych, wiadomości są co 1 sekundę renderowanie na oddzielnej stronie i przesyłane na chat za pomocą JavaScript’u. Powoduje to, że możliwe jest pisanie w czasie ”rzeczywistym”, bez potrzeby ciągłego odświeżania strony. Ten problem został rozwiązany w bibliotece Socket.IO, która umożliwia asynchroniczne przesyłanie danych w czasie rzeczywistym. Z racji na to, że jest to projekt z gniazd a nie web development’u, zdecydowaliśmy się jej nie używać. Stworzyło to problem z synchronizacją pobierania wiadomości, oraz reszty informacji przesyłanych pomiędzy serwerem i klientem. Zostało to rozwiązane za pomocą zmiennej globalnej która informuję czy nie są wysyłane lub pobierane inne dane przez resztę aplikacji.

6 Dodawanie znajomych

Po przejściu do okna dodawania znajomych użytkownik widzi pole do wpisania nazwy użytkownika i przycisk zatwierdzający. Dane są przesyłane do serwera, który sprawdza czy podana nazwa istniej, czy nie jest już znajomych użytkownika, oraz czy użytkownik nie podał sam siebie. Jeżeli nie, to dodaje do znajomego do bazy danych i przesyła stosowny komunikat. Aplikacja jeżeli wszystko jest w porządku przenosi użytkownika z powrotem na chat. Jeżeli użytkownik dodał znajomego może do niego pisać. Nie oznacza to jednak, że drugi zobaczy te wiadomości. Żeby to się stało obaj użytkownicy muszą dodać się nawzajem. Rozwiązuje to problem potwierdzania znajomości.

7 Testy

W czasie pracy nad aplikacją były prowadzone testy sprawdzające jej stabilność, oraz działanie. Mimo, że aplikacja jest całkiem stabilna, to ciągłe ściąganie wiadomości czasami powoduje błędy (szczególnie gdy mamy odpalone wiele okien chatu na raz, co powoduję desynchronizację procesu) W przypadku błędu aplikacja próbuję się rozłączyć z serwerem, następnie uruchamia skrypt Bashowy, który restartuję aplikację. W takim przypadku użytkownik na chwile straci chat, który powinien wrócić po odświeżeniu strony. Proces ten działa raczej bez problemów w środowisku Linux. Aby zadziałał na Windowsie niezbędne jest poprawne zdefiniowanie ścieżki do Pythona i FLaska, aby skrypt rozumiał, że ma włączyć aplikację Flaskową. Połączenie się dwóch komputerów również było testowane i działa poprawnie (testowane dla dwóch komputerów z windowsem 10). Działanie na Linuxie było testowane tylko lokalnie na masynie wirtualnej udało się połączyć Ubuntu 19.10 z Debianem 10.